Optical Parallel Quaternary Signed Digit Multiplier For Large Scale Two-Dimensional Array Using Digit-Decomposition Plane Representation

Alaa A. W. Al-Saffar
Basrah Technical Institute
Department of Electronics
Basrah-Iraq
E-mail: alaaasaffar@yahoo.com

Abstract

An optical parallel quaternary signed digit (QSD) two-dimensional array multiplier based on digit-decomposition (DDP) representation and duplication-shifting-superimposing algorithm is proposed in this paper. The multiplication operation is done in three steps; one for partial products generation and the other two steps perform accumulation to find the DDP planes of the final result array. QSD multiplication and addition rules are used to obtain a newly derived equations which are suitable for easy optical implementation using basic optical tools. Finally, simulation results are presented to validate the successful of the multiplication operation.

الضارب الضوئي المتوازي لكتلة كبيرة ثنائية الابعاد للارقام المؤشرة الرباعية باستخدام تمثيل مستويات فصل الأرقام

د. علاء عبالوهاب الصفار
المعهد التقني بالبصرة
قسم الالكترونيك
بصرة - العراق

الملخص

تقدم هذه الورقة تصميما ضوئيا متوازيا لإجراء عملية الضرب لكتلة ثنائية الأبعاد من الأرقام المؤشرة الرباعية باستخدام تمثيل مستويات فصل الأرقام و على خوارزمية مضاعفة - زحف- إضافة الأرقام. إن عملية الضرب تتم في ثلاث خطوات, يتم في الخطوة الأولى الحصول على نواتج الضرب الجزئي بينما يتم في الخطوتين اللاحقتين جمع هذه النواتج لغرض الحصول على الناتج النهائي. تم استخدام جداول الضرب و الإضافة للأعداد المؤشرة الرباعية لغرض اشتقاق معادلات جديدة مناسبة للتمثيل الضوئي باستخدام الأدوات التقليدية. وأخيرا, نتائج المحاكاة المعروضة تثبت نجاح عملية الضرب.

1. Introduction

To fulfill the theories of the new technologies that require very high speed, there is a need to ultrahigh speed processing devices. Optical parallel processing has been used by the researchers for data processing and used this technology in various purposes such as image processing, data computing, control systems, etc [1,2]. The use of light in the optical processing systems leaded to many advantages such as high speed and the parallelism [3]. Also in these systems, the light signals can pass through each other without interacting or influencing the data carried by.

Digital computer systems are suffered from the carry propagation that might appear in the intermediate steps of arithmetic operations because of its sequential operation. The carry propagation made the processing time depends on the length of the numbers under processing. This problem was solved by proposing many optical techniques which performed the parallel arithmetic operations in constant processing time independent of the length of numbers. These optical computing techniques were established using many parallel algorithms in order to perform the arithmetic operations [4,5]. One of a common technique used in optical parallel arithmetic operations is the ***"Signed-Digit number systems"*** (SD) [6]. Also, parallel optical logic gates had been suggested to process large amounts of data in parallel and in speed of light [7]. In SD arithmetic, many works have been suggested for manipulating the addition and multiplication processes based on various optical schemes for processing SD data in parallel. A parallel optical modified SD (MSD) two-dimensional (2D) array multiplier using digit-decomposition plane (DDP) representation method has suggested in [8]. Also, a synthetic correlation-based parallel trinary SD (TSD) multiplier has designed in [9]. Recently, a parallel QSD multiplier using symbolic substitution (SS) technique is proposed in [10].

In this paper, parallel optical 2D array multiplier for quaternary SD (QSD) number is suggested. The process is based on DDP representation technique. A newly derived equations have been derived which are suitable for optical implementation using basic optical tools. Finally, a simulation example is introduced to verify the multiplication operation.

2. SD Numbers

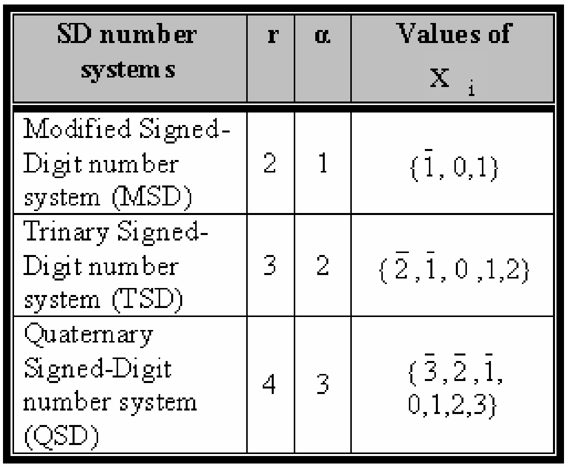
The decimal number can be represented in SD number form as:

$$D = \sum_{i=0}^{n-1} x_i \, r^i \qquad (1)$$

where D is the decimal number, x_i is the i-th digit of SD number, $x_i \in \{-\alpha , -(\alpha-1) , \ldots , (\alpha -1) , \alpha \}$, $\alpha = r-1$, r is the radix of SD number system, and n is the number of digits in SD number.

The common used values of r are {2, 3, and 4}. Table (1) shows the three SD number systems which had been widely used. Note that, -1, -2, and -3 will be denoted by $\bar{1}$, $\bar{2}$, and $\bar{3}$, respectively.

Table (1): The three SD number systms.



SD number systems	r	α	Values of x_i
Modified Signed-Digit number system (MSD)	2	1	$\{\bar{1}, 0, 1\}$
Trinary Signed-Digit number system (TSD)	3	2	$\{\bar{2}, \bar{1}, 0, 1, 2\}$
Quaternary Signed-Digit number system (QSD)	4	3	$\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$

The redundancy feature of SD numbers made it very powerful to implement an optical parallel arithmetic unit with carry-free or carry-limited processing [11]. Note that a higher radix r of the SD number system gives a large range of numbers with fewer SD digits comparing with the binary number system that will require a large number of binary digits to offer a large range of binary numbers].

3. DDP Optical Coding Methods

The SD numbers are represented in different methods. The coding methods depend on pixel assignment to the SD numbers. Mostly, two values of transparencies, opaque (dark) and transparent (white), are assigned in order to code the SD numbers by special arrangement of these two pixels.

DDP representation has proposed by *Hongxin Huang* and *et.al.* [8]. It is an extension for bit-plane representation method [12]. DDP representation can be applied to code large 2D data arrays of SD numbers.

In this paper, an expansion for the MSD DDP coding method performed in Ref [9] has been successfully applied to implement QSD number. The proposed scheme uses 7 DDP planes (DDP $3,2,1,0,\bar{1},\bar{2},\bar{3}$). Figure 1 shows example for QSD data arrays coded by DDP scheme. It is noted that the coding is done according to the following rule, if the (i,j)-th digit in the 2D QSD numbers array equal to 0 (1, 2 , 3 , $\bar{1}$, $\bar{2}$, and$\bar{3}$), then the corresponding (i,j)-th pixel of the DDP-0 (1, 2 , 3 , $\bar{1}$, $\bar{2}$, and$\bar{3}$)plane will be transparent (white) and the corresponding (i,j)-th pixels of all the other planes will be opaque (dark).

$$\begin{bmatrix} 7 \\ 237 \\ -108 \\ -138 \end{bmatrix}_{10} = \begin{bmatrix} 0013 \\ 330\bar{3} \\ \bar{1}\bar{2}\bar{3}0 \\ \bar{3}32\bar{2} \end{bmatrix}_{QSD}$$

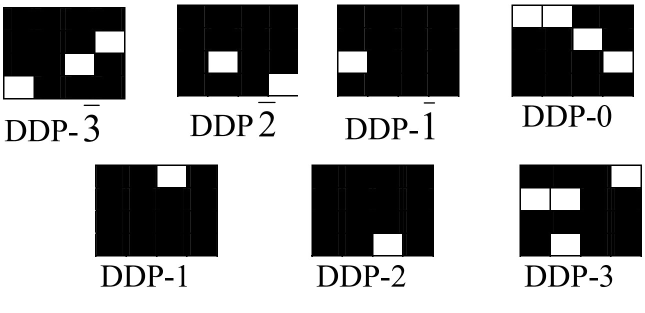


Fig. (1) : DDP representations of 2D QSD data arrays.

4. Multiplication Scheme

The multiplication process is based on DDP representation technique and the duplication-shifting-superimposing (DSS) multiplication algorithm [13]. Generally, the SD multiplication operation takes in two parts. The first part is the optical system that generates the DDP planes of the partial product array in parallel. In this part, logical formulas that represent the SD multiplier rules will be derived depending on the parallel SD multiplication rules. Then, these logical formulas are combined with the DSS algorithm principles to form n channels operating in parallel to generate n partial product arrays, each one as DDP planes, simultaneously. In the second part, an accumulation operation of the n partial product arrays to find the DDP planes of the final result array is performed. Therefore, an optical parallel two-step QSD array adder is used as tree adders.

4.1 QSD Multipliers System Operation

Two M×N×n SD data arrays A (multiplicand) and B (multiplier) are coded as DDP planes. These DDP planes form the logical formulas of the SD multipliers in the n channels of the DSS algorithm. Each channel generates one partial product array PP_k, and all channels are operated in parallel. So, a total of n PP_k s are generated simultaneously. The whole system operation (see Fig. 2) is described in the following steps:

1. Double each pixel in the DDP planes of A and B arrays horizontally to be M×N×2n DDP planes.
2. Replicate the doubled DDP planes, which are used in the logical formulas of the SD multiplier, n times but with proper shift operation explained in step 3.
3. The doubled DDP planes of the k-th channel, where $k = \{-\frac{n}{2},\cdots,-1,0,1,\cdots,\frac{n}{2}-1\}$ if

n is even number, and $k = \{-\frac{n-1}{2}, \cdots, -1, 0, 1, \cdots, \frac{n-1}{2}\}$ if n is odd number are shifted according to the following two conditions:

a) If $k \geq 0$, then each number of the doubled DDP planes of array A will be shifted 2k positions to the left, and each number of the doubled DDP planes of array B are shifted 2k+1 positions to the right.

b) Conversely, If k<0, then each number of the doubled DDP planes of array A are shifted $|2k|$ positions to the right, and each number of the doubled DDP planes of array B will be shifted $|2k+1|$ positions to the left.

4. Now, these doubled and shifted DDP planes are used to construct the logical formulas of the SD multiplier. The SD multiplier has n channels, the k-th channel are used the logical formulas with doubled and shifted DDP planes that are shifted according to k.

5. The M×N×2n DDP planes of the n partial product arrays PP_ks are generated, in parallel, by n channels. In order to accumulate these n PP_ks, a parallel two-step SD adders with DDP representation are used as the tree adders. Each tree adder consists of n-1 adders distributed to $O(\log_2 n)$ stages to generate the final result array Z, where $O(\ )$ denotes the rounding function.

From the experimental results, the final result array Z will be represented as $M \times N \times (2n+\beta)$ DDP planes where the symbol β represents a numerical value $\{0,1,2\ldots\}$, and it is increased depending on the number of the DSS channels. In other word, β depends on the maximum number of SD digits in the SD numbers.

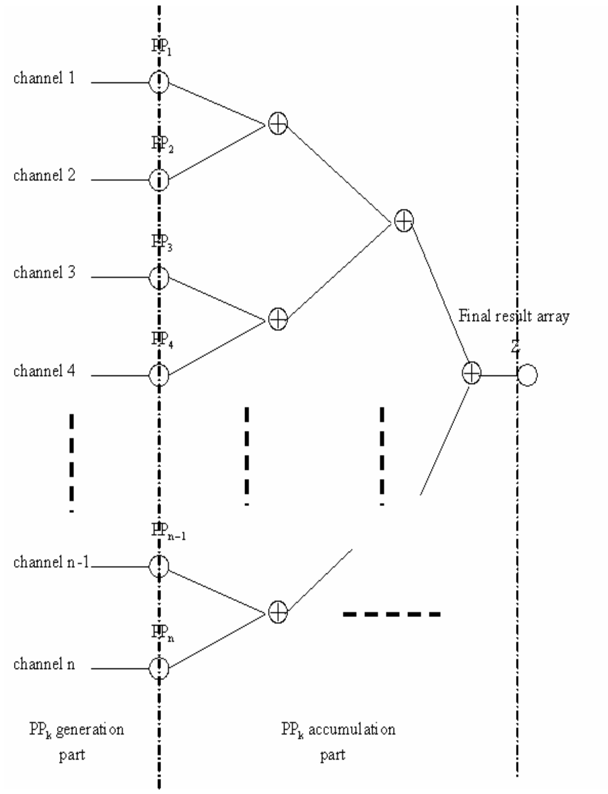


Fig.(2): The system operation of the QSD multiplier.

The computational rules are derived as follows:

- The first step is to generate the intermediate product p_i and the intermediate carry u_i using Table(2) [4].

$$\begin{aligned} IF\ (x_i, y_i) = {} & (3,3)\ \text{OR}\ (\bar{3},\bar{3})\ \text{OR}\ (1,1) \\ & \text{OR}\ (\bar{1},\bar{1})\ \text{OR}\ (3,\bar{1})\ \text{OR}\ (\bar{3},1) \\ & \text{OR}\ (\bar{1},3)\ \text{OR}\ (1,\bar{3}) \\ & \text{THEN}\ p_i = 1 \end{aligned} \tag{2}$$

So, the following rule can be set as a logical rule in generating the P1 DDP plane:

$$P1 = A3 * B3 + A\bar{3} * B\bar{3} + A1 * B1 + A\bar{1} * B\bar{1}$$
$$+ A3 * B\bar{1} + A\bar{3} * B1 + A\bar{1} * B3 + A1 * B\bar{3}$$

(3)

Rule (3) can be rewritten as:

$$P1 = (A3 + A\bar{1}) * (B3 + B\bar{1})$$
$$+ (A1 + A\bar{3}) * (B1 + B\bar{3})$$

(4)

- The rest DDP planes of the intermediate product array (P2, P0, $P\bar{1}$, and $P\bar{2}$) can be generated from Table (2) for the following rules which are newly derived:

$$P2 = A2 * (B3 + B1) + A\bar{2} * (B\bar{1} + B\bar{3}) +$$
$$(A3 + A1) * B2 + (A\bar{1} + A\bar{3}) * B\bar{2}$$
$$P0 = A0 + \overline{A0} * B0 +$$
$$(A2 + A\bar{2}) * (B2 + B\bar{2})$$
$$P\bar{1} = (A3 + A\bar{1}) * (B1 + B\bar{3})$$
$$+ (A1 + A\bar{3}) * (B3 + B\bar{1})$$
$$P\bar{2} = A2 * (B\bar{1} + B\bar{3}) + A\bar{2} * (B3 + B1)$$
$$+ (A\bar{1} + A\bar{3}) * B2 + (A3 + A1) * B\bar{2}$$

(5)

Also, from Table (2), the DDP planes of the intermediate carry array U (U2, U1, U0, $U\bar{1}$, and $U\bar{2}$) can be obtained using the following rules:

$$U2 = A3 * B3 + A\bar{3} * B\bar{3}$$
$$U1 = A3 * (B2 + B1) + A\bar{3} * (B\bar{1} + B\bar{2})$$
$$+ A2 * (B3 + B2) + A\bar{2} * (B\bar{2} + B\bar{3})$$
$$+ A1 * B3 + A\bar{1} * B\bar{3}$$
$$U0 = A0 + \overline{A0} * B0 + (A2 + A\bar{2}) * (B1 + B\bar{1})$$
$$+ (A1 + A\bar{1}) * (B2 + B1 + B\bar{1} + B\bar{2})$$
$$U\bar{1} = A3 * (B\bar{1} + B\bar{2}) + A\bar{3} * (B2 + B1)$$
$$+ A2 * (B\bar{2} + B\bar{3}) + A\bar{2} * (B3 + B2)$$
$$+ A1 * B\bar{3} + A\bar{1} * B3$$
$$U\bar{2} = A3 * B\bar{3} + A\bar{3} * B3$$

(6)

Table (2): QSD multiplication rules to generate p_i and u_i .

groups	(multiplicand,multiplier)	intermediate results	
G_i	(x_i, y_i)	p_i	u_i
G_1	$(3,3),(\bar{3},\bar{3})$	1	2
G_2	$(3,2),(2,3),(\bar{3},\bar{2}),(\bar{2},\bar{3})$	2	1
G_3	$(2,2),(\bar{2},\bar{2})$	0	1
G_4	$(3,1),(1,3),(\bar{3},\bar{1}),(\bar{1},\bar{3})$	$\bar{1}$	1
G_5	$(2,1),(1,2),(\bar{2},\bar{1}),(\bar{1},\bar{2})$	2	0
G_6	$(1,1),(\bar{1},\bar{1})$	1	0
G_7	$(0,1),(0,2),(0,3),(0,\bar{1}),(0,\bar{2}),(0,\bar{3}),(0,0),$ $(1,0),(2,0),(3,0),(\bar{1},0),(\bar{2},0),(\bar{3},0)$	0	0
G_8	$(1,\bar{1}),(\bar{1},1)$	$\bar{1}$	0
G_9	$(\bar{2},1),(1,\bar{2}),(\bar{1},2),(2,\bar{1})$	$\bar{2}$	0
G_{10}	$(3,\bar{1}),(\bar{1},3),(\bar{3},1),(1,\bar{3})$	1	$\bar{1}$
G_{11}	$(2,\bar{2}),(\bar{2},2)$	0	$\bar{1}$
G_{12}	$(3,\bar{2}),(\bar{2},3),(\bar{3},2),(2,\bar{3})$	$\bar{2}$	$\bar{1}$
G_{13}	$(3,\bar{3}),(\bar{3},3)$	$\bar{1}$	$\bar{2}$

- The second step is to add p_i and u_i to get the intermediate sum w_i and intermediate carry T_i according to the rules set at Table (3). This step has eight logical formulas as shown in eq. 6. Five of these equations are for generating the DDP planes of the intermediate sum array w (W2, W1, W0, $W\bar{1}$, and $W\bar{2}$) and the others three are for generating the DDP planes of the intermediate carry array T ($T1$, T0, and $T\bar{1}$) .

$$W2 = P2 * U'0 + P0 * U'2 + P1 * U'1$$
$$W1 = P\bar{1} * (U'2 + U'\bar{2}) + (P2 + P\bar{2}) * U'\bar{1} + P0 * U'1 + P1 * U'0$$
$$W0 = (P2 + P\bar{2}) * (U'2 + U'\bar{2}) + P0 * U'0 + P1 * U'\bar{1} + P\bar{1} * U'1$$

$$W\bar{1} = P1 * (U'2 + U'\bar{2}) + (P2 + P\bar{2}) * U'1 + P0 * U'\bar{1} + P\bar{1} * U'0$$

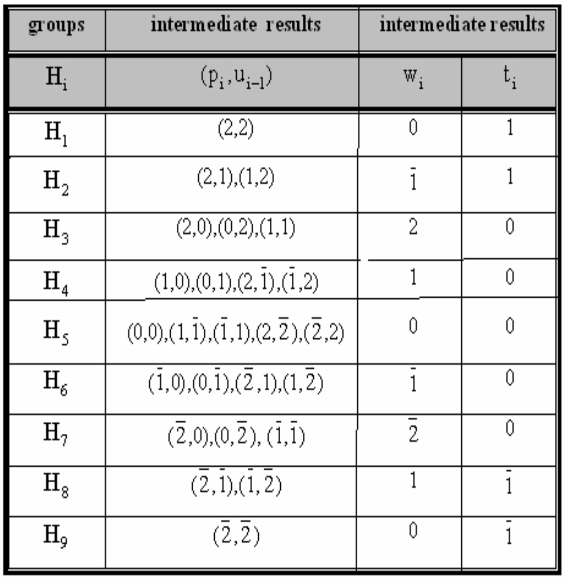
$$W\bar{2} = P\bar{2} * U'0 + P0 * U'\bar{2} + P\bar{1} * U'\bar{1}$$
$$T1 = P2 * (U'2 + U'1) + P1 * U'2$$

$$\begin{aligned} T0 = {} & (P2 + P1) * (U'0 + U'\bar{1} + U'\bar{2}) + (P\bar{2} + P\bar{1}) * (U'0 + U'1 + U'2) \\ & + P1 * U'1 + P\bar{1} * U'\bar{1} + P0 \end{aligned}$$

$$T\bar{1} = P\bar{2} * (U'\bar{2} + U'\bar{1}) + P\bar{1} * U'\bar{2} \qquad (7)$$

The symbol $(\,'\,)$ on the right top of the U refers to the shifting one position to the left for each number included in the planes.

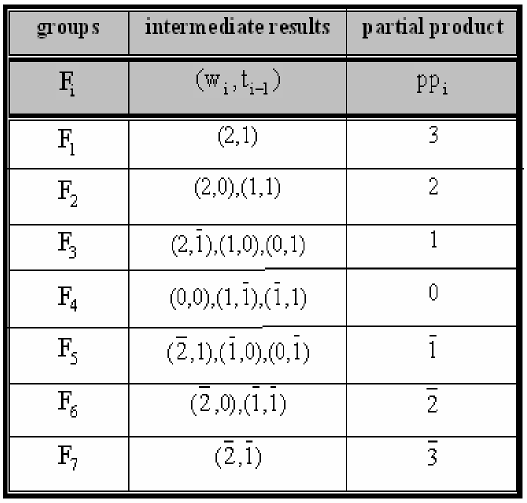
Table (3): QSD addition rules to generate w_i and T_i .



groups	intermediate results	intermediate results	
H_i	(p_i, u_{i-1})	w_i	t_i
H_1	(2,2)	0	1
H_2	(2,1),(1,2)	$\bar{1}$	1
H_3	(2,0),(0,2),(1,1)	2	0
H_4	$(1,0),(0,1),(2,\bar{1}),(\bar{1},2)$	1	0
H_5	$(0,0),(1,\bar{1}),(\bar{1},1),(2,\bar{2}),(\bar{2},2)$	0	0
H_6	$(\bar{1},0),(0,\bar{1}),(\bar{2},1),(1,\bar{2})$	$\bar{1}$	0
H_7	$(\bar{2},0),(0,\bar{2}), (\bar{1},\bar{1})$	$\bar{2}$	0
H_8	$(\bar{2},\bar{1}),(\bar{1},\bar{2})$	1	$\bar{1}$
H_9	$(\bar{2},\bar{2})$	0	$\bar{1}$

- During the third-step w_i and T_i are added to generate the partial product pp_i according to the following newly derived equation using Table 4.

Table (4) :. QSD addition rules to generate pp_i .



groups	intermediate results	partial product
F_i	(w_i, t_{i-1})	pp_i
F_1	(2,1)	3
F_2	(2,0),(1,1)	2
F_3	$(2,\bar{1}),(1,0),(0,1)$	1
F_4	$(0,0),(1,\bar{1}),(\bar{1},1)$	0
F_5	$(\bar{2},1),(\bar{1},0),(0,\bar{1})$	$\bar{1}$
F_6	$(\bar{2},0),(\bar{1},\bar{1})$	$\bar{2}$
F_7	$(\bar{2},\bar{1})$	$\bar{3}$

$$PP3 = W2 + T'1$$
$$PP2 = W2 * T'0 + W1 * T'1$$

$$PP1 = W2 * T'\bar{1} + W1 * T'0 + W0 * T'1$$

$$PP0 = W0 * T'0 + W1 * T'\bar{1} + W\bar{1} * T'1$$

$$PP\bar{1} = W\bar{2} * T'1 + W\bar{1} * T'0 + W0 * T'\bar{1}$$

$$PP\bar{2} = W\bar{2} * T'0 + W\bar{1} * T'\bar{1}$$

$$PP\bar{3} = W\bar{2} * T'\bar{1} \qquad (8)$$

At this moment, $7\times n$ $M\times N\times 2n$ DDP planes are already presented at the end of the n channels. The n partial product arrays pp_k can be accumulated to obtain the final result array Z as seven $M\times N\times(2n+\beta)$ DDP planes Z3, Z2, Z1, Z0, $Z\bar{1}$, $Z\bar{2}$, and $Z\bar{3}$.

5. Optical Implementation

Figure (3) presents the optical implementation of the QSD array multiplier with the DDP plane representation method using basic optical tools such as: beam combiner (BC), beam splitter BS), mirrors, and spatial light modulators [14,15]. In the first step, the intermediate product p_i and intermediate carry u_i are generated. These parameters are used in the second step which performs accumulation to generate the intermediate sum W_i and carry T_i.

To simplify the optical implementation, one of the parameters in rules 4, 5 , and 6 is rewritten in terms of the other parameters as shown in eq. 9.

$$
\begin{aligned}
P\bar{2} &= \overline{P2+P1+P0+P\bar{1}} \\
U\bar{1} &= \overline{U2+U1+U0+U\bar{2}} \\
W0 &= \overline{W2+W1+W\bar{1}+W\bar{2}} \\
PP0 &= \overline{PP3+PP2+PP1+PP\bar{1}+PP\bar{2}+PP\bar{3}}
\end{aligned}
\qquad (9)
$$

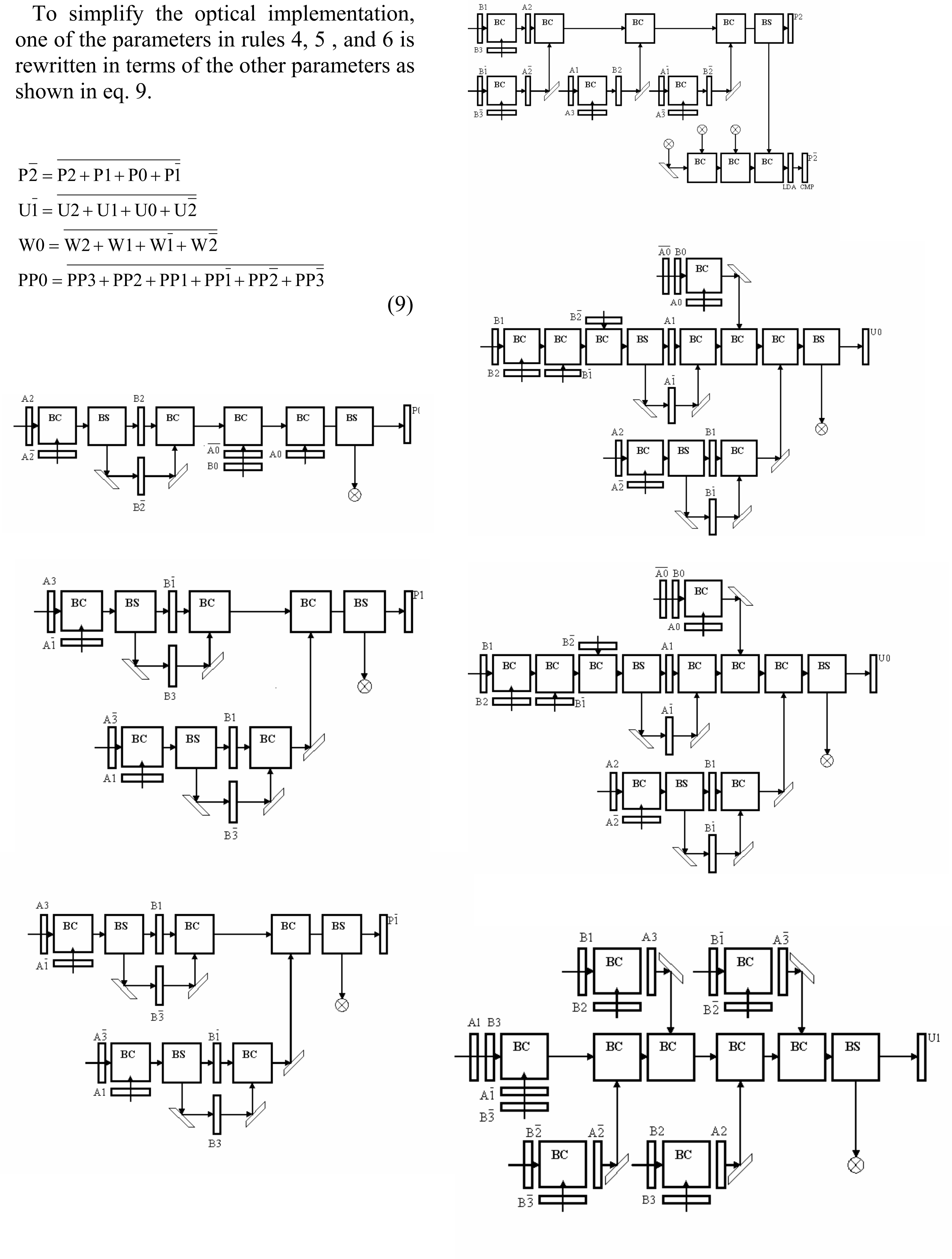


Fig. (3) : First step optical implementation of QSD multiplier.

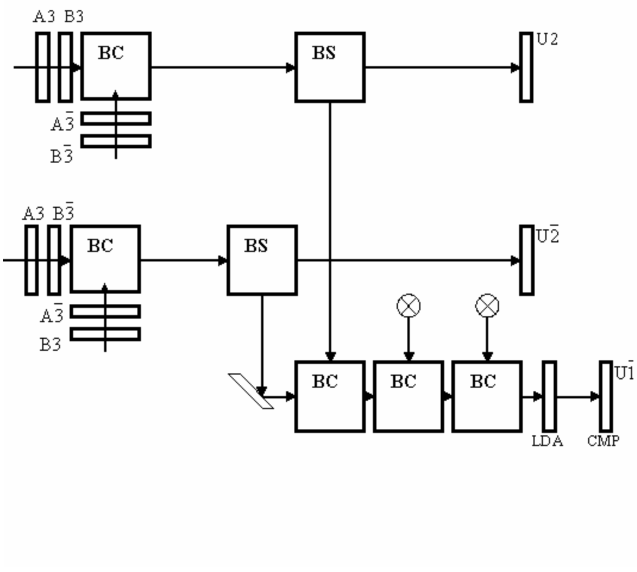


Fig. (3): continued.

Figure (4) shows the third step of the multiplier that generates the product pp_i .

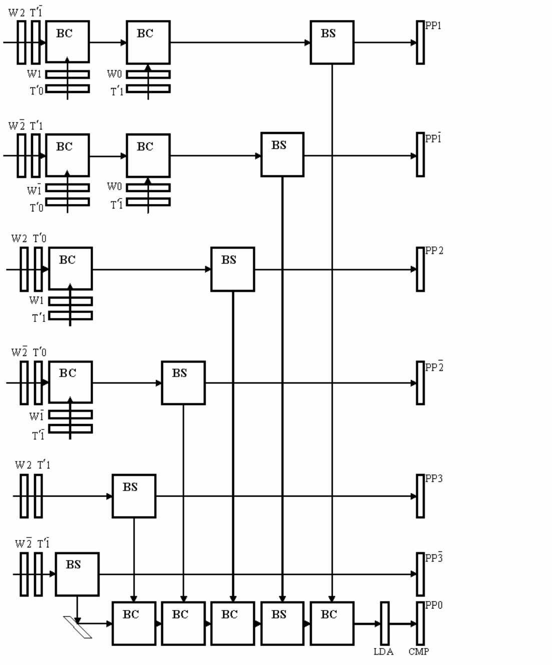


Fig. (4) : Third step optical implementation of QSD multiplier

4.2 Simulation Results

Simulation computer programs have been built using C++ language to test the parallel optical QSD array multipliers based on DDP representation. Two $10\times2\times4$ QSD arrays A and B will be used to test the proposed QSD array multiplier.

$$A=\begin{bmatrix} 255 & -255 \\ 94 & 6 \\ 135 & 72 \\ -210 & -140 \\ 41 & 200 \\ 105 & -100 \\ 197 & 10 \\ -237 & 25 \\ 55 & -113 \\ 255 & -255 \end{bmatrix}_{10}=\begin{bmatrix} 3333 & \bar{3}\bar{3}\bar{3}\bar{3} \\ 1132 & 0012 \\ 2013 & 1020 \\ \bar{3}\bar{1}0\bar{2} & \bar{2}0\bar{3}0 \\ 0221 & 3020 \\ 1221 & \bar{1}\bar{2}\bar{1}0 \\ 3011 & 0022 \\ \bar{3}\bar{2}\bar{3}\bar{1} & 0121 \\ 0313 & \bar{1}\bar{3}0\bar{1} \\ 3333 & \bar{3}\bar{3}\bar{3}\bar{3} \end{bmatrix}_{QSD}$$

$$B=\begin{bmatrix} 255 & 255 \\ 112 & 31 \\ 41 & 174 \\ 87 & 200 \\ -127 & -10 \\ -205 & -210 \\ 14 & -199 \\ -50 & 125 \\ 11 & -68 \\ -255 & -255 \end{bmatrix}_{10}=\begin{bmatrix} 3333 & 3333 \\ 1300 & 0133 \\ 0221 & 2232 \\ 1113 & 3020 \\ \bar{1}\bar{3}\bar{3}\bar{3} & 00\bar{2}\bar{2} \\ \bar{3}0\bar{3}\bar{1} & \bar{3}\bar{1}0\bar{2} \\ 0032 & \bar{3}0\bar{1}\bar{3} \\ 0\bar{3}0\bar{2} & 1331 \\ 0023 & \bar{1}0\bar{1}0 \\ \bar{3}\bar{3}\bar{3}\bar{3} & \bar{3}\bar{3}\bar{3}\bar{3} \end{bmatrix}_{QSD}$$

The doubled and shifted DDP planes of the QSD arrays A and B for the four DSS channels {-2, -1, 0, 1} are shown in Fig.5-a. In Fig.5.b the DDP planes of the intermediate product P and shifted intermediate carry U' arrays, which are generated in four DSS channels simultaneously, are presented. These P and U' DDP planes will enter the next step of the PP generation, which are processed in parallel to produce the DDP planes of the intermediate sum W, and the shifted intermediate carry T'arrays in the four DSS channels. The W and T' DDP planes are illustrated in Fig.5c. Using these W and T' DDP planes in the third step of the PP_k generation part, the four QSD partial product arrays PP can be generated. Each one of the PP arrays has seven $10\times2\times8$ DDP planes. Here, a single zero (β=1) must be padded in the MSB positions of each number included in the PP arrays to expand the DDP planes to $10\times2\times9$ pixel resolution. Figure 5d shows the four PP arrays as $10\times2\times9$ DDP planes. At this moment, we have four PP arrays, each coded in seven$10\times2\times9$ DDP planes,

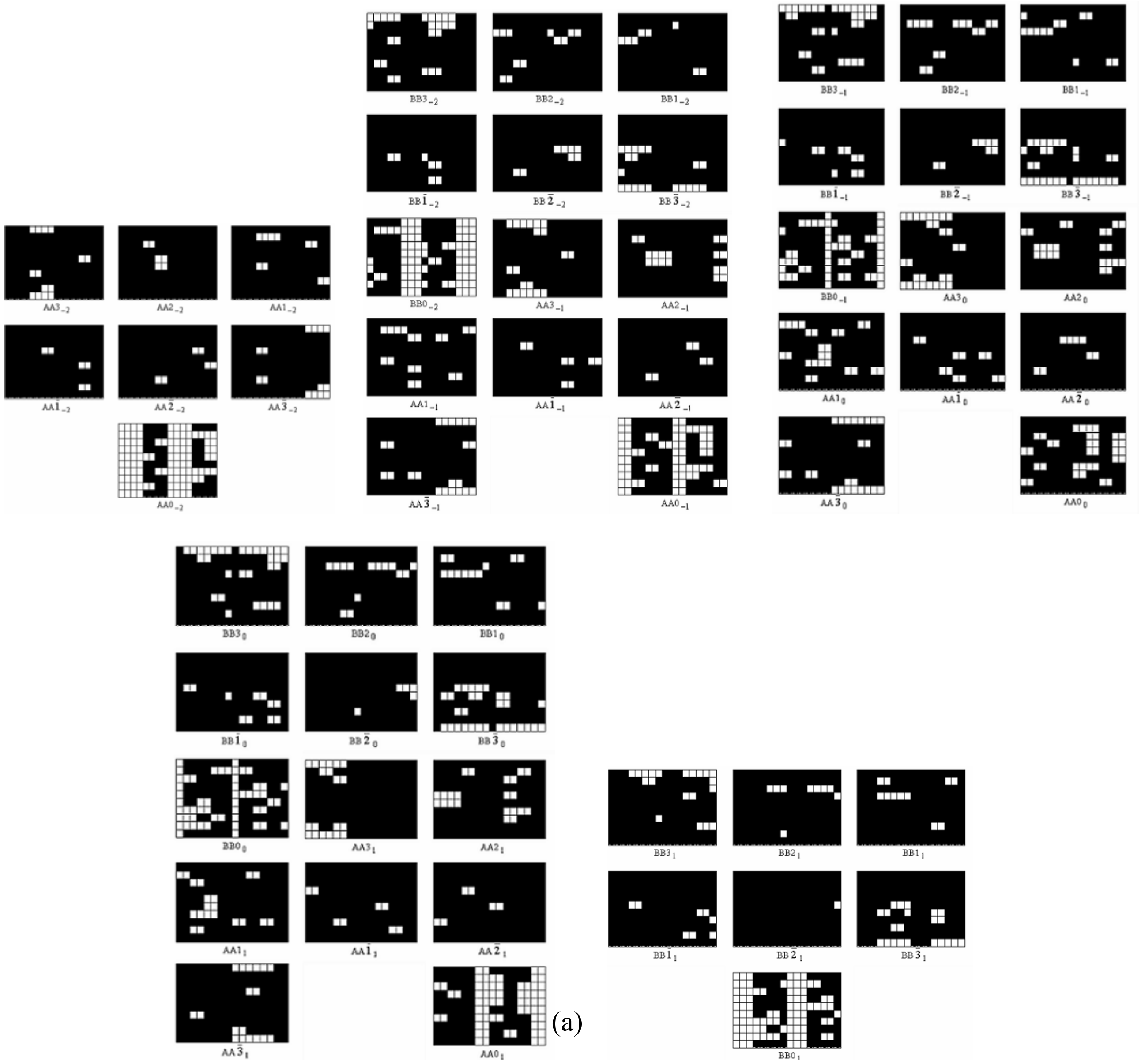


Fig.(5): Simulation results of the parallel optical QSD array multiplier.

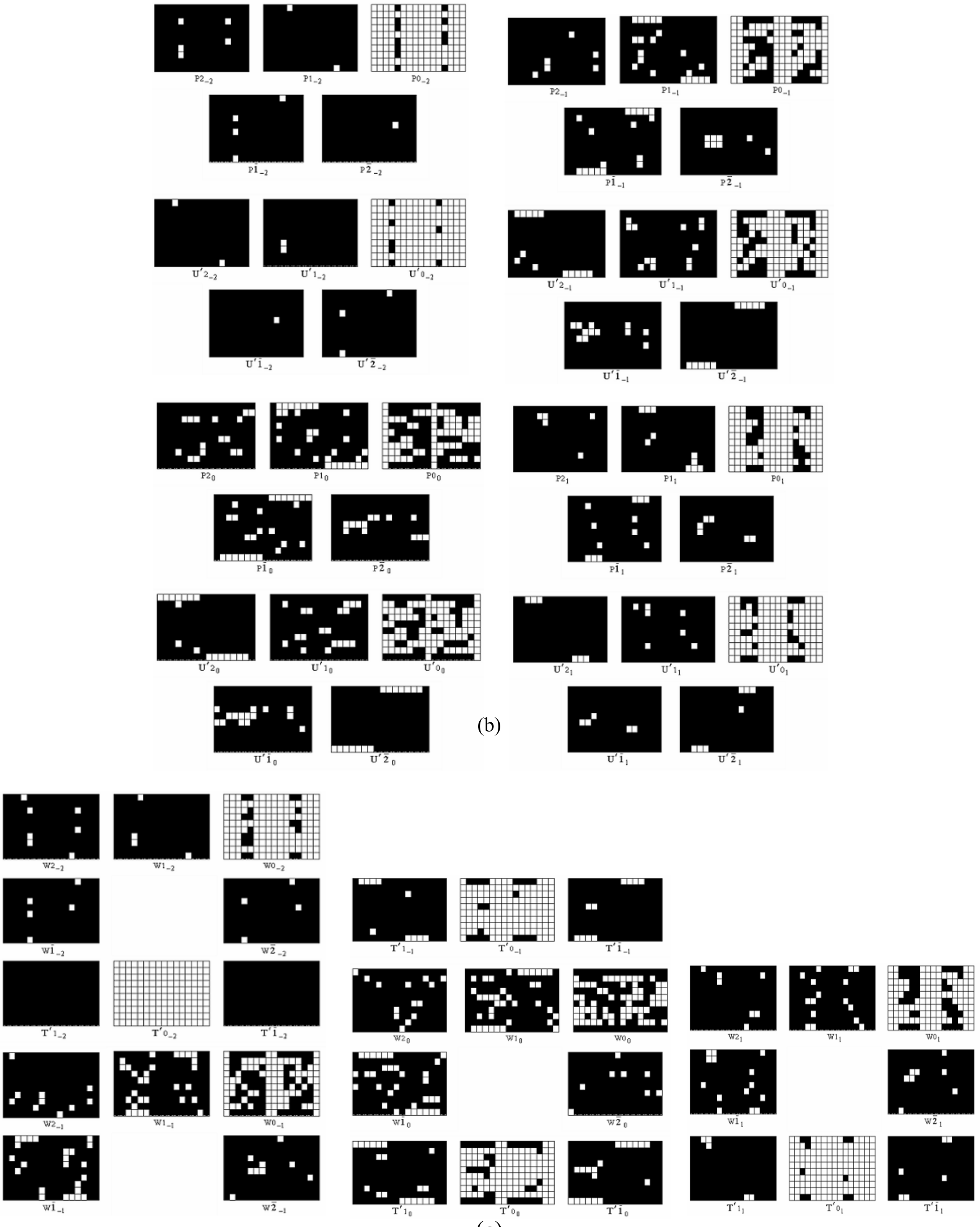
(a): The four doubled and shifted versions of the DDP planes of the QSD arrays A and B.

(b): DDP planes of the four intermediate partial P arrays and the shifted intermediate carry U' arrays.

(c): DDP planes of the four intermediate sum W arrays and the shifted intermediate carry T' arrays.

(d): DDP planes of the four generated partial product arrays PPs.

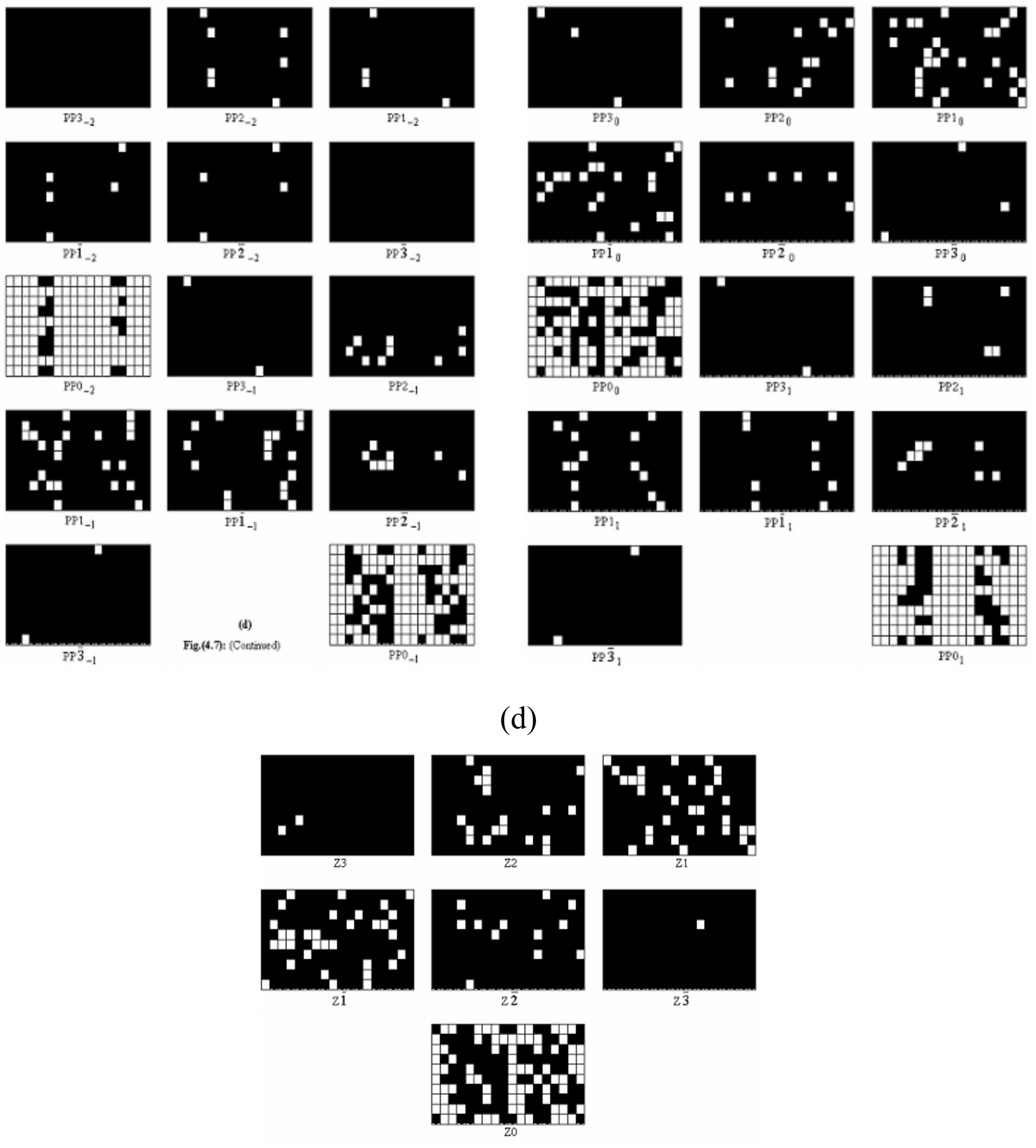
(e): DDP planes of the final results QSD array Z.



(b)

(c)

Fig. (5): Continued



(d)

(e)

Fig. (5): Continued.

produced by the four DSS channels as shown in Fig.5e.

When these seven final result array $10\times2\times9$ DDP planes are decoded and composed, the $10\times2\times9$ QSD array Z which includes the final results is obtained as present below. The single zero that has been padded as $\beta=1$ to each DDP planes of the partial product arrays PP is occupied now by 1 and $\bar{1}$ in 65025 and -65025 number in the final result QSD array Z.

$$Z=\begin{bmatrix} 100\bar{1}20001 & \bar{1}001\bar{2}000\bar{1} \\ 01\bar{1}\bar{2}10200 & 00001\bar{1}0\bar{2}2 \\ 00111220\bar{1} & 01\bar{1}010\bar{1}00 \\ 0\bar{1}0\bar{2}1\bar{2}21\bar{2} & 0\bar{1}\bar{3}\bar{1}\bar{1}\bar{1}200 \\ 00\bar{1}\bar{1}0\bar{1}\bar{1}\bar{2}1 & 000\bar{2}01\bar{1}00 \\ 0\bar{1}\bar{1}\bar{1}00\bar{1}\bar{1}\bar{1} & 011020020 \\ 00023\bar{1}012 & 000\bar{2}010\bar{1}\bar{2} \\ 003\bar{1}21022 & 001\bar{1}01\bar{1}11 \\ 0000212\bar{1}1 & 002\bar{1}20010 \\ \bar{1}001\bar{2}000\bar{1} & 100\bar{1}20001 \end{bmatrix}_{QSD}=\begin{bmatrix} 65025 & -65025 \\ 10528 & 186 \\ 5535 & 12528 \\ -18270 & -28000 \\ -5207 & -2000 \\ -21525 & 21000 \\ 2758 & -1990 \\ 11850 & 3125 \\ 605 & 7684 \\ -65025 & 65025 \end{bmatrix}_{10}$$

One can expect that the throughput (thput) of the optical parallel SD array multiplier can be calculated by the following equation [5]:

$$thput=\frac{M\times N\times n}{(2\times O(\log_2 n)+r-1)\times(\mathrm{SLM}_\tau+2\tau)}$$

where the term $2\times O(\log_2 n)$ determines the number of steps in the tree adder and the term (r-1) determines the number of steps in the partial product generation part. With assuming that τ=20ns, using SLMs with 5000×5000 pixel resolution and 50 μs response time specifications, we can get 38.4 Gbps QSD multiplier, all for multiplying 390000 pairs of 32-digit SD numbers simultaneously.

6. Conclusions

An optical design for parallel array QSD multipliers is presented. The proposed SD multipliers combine the DDP representation, parallel SD multiplication algorithms, and the DSS multiplication algorithm. The SD multiplier consists of two parts, the partial products generation and the partial products accumulation. Tree QSD adders are built to accumulate the partial products.

References

[1] A. S. Awwal and K. M. Iftekharuddin," Graphical Approach for Multiple Valued Logic Minimization," *Opt.Eng*., Vol.38, No.3, pp.462-467, March 1999.

[2] B. G. Boone," Signal Processing Using Optics," *Oxford University Press*, 1998.

[3] S. Kawaai, Y. Tashiro, H. Ichinose, and K.Kasahara," Cascade-Connective Optical Parallel Logic Processor," App. Opt., Vol. 31, No.2, pp. 178-185, Jan, 1992.

[4] S. Zhang and M. A. Karim," Optical," Symmetrically Recoded Quaternary Signed-Digit Arithmetic Using Arithmetic Processing Using Improved Redundant Binary Algorithm," *Opt.Eng*., Vol.38, No.3, pp.415-421, March 1999.

[5] A. K. Cherri," Signed-Digit Arithmetic for Optical Computing: Digit Groping and Pixel Assignment for Spatial Coding," *Opt.Eng*., Vol.38, No.3, pp.422-431, March 1999.

[6] M. S. Alam, A. K. Cherri, and A. Chatterjeaa Shared Content-Addressable Memory," *Opt.Eng*., Vol.35, No.4, pp.1141-1149, Apr. 1996.

[7] S. Zhang and M. A. Karim," Programmable Modified Signed-Digit Addition Module Based on Binary Logic Gates," *Opt.Eng*., Vol.38, No.3, pp.456-461, March 1999.

[8] H. Huang, M. Itoh, and T. Yatagai," Optical Scalable Parallel Modified Signed-Digit Algorithms for Large Scale Array Addition and Multiplication Using Digit-Decomposition-Plane Representation," *Opt.Eng*., Vol.38, No.3, pp.432-440, March 1999.

[9] F. Ahmmed, A. S. Awwal, and G. J. Power," Synthetic Correlation Modified Signed-Digit Processing," *Opt.Eng*., Vol.38, No.3, pp.449-455, March 1999.

[10] A. K. Cherri and M. S. Alam," Parallel Computation of Complex Elementary Functions Using Quaternary Signed-Digit Arithmetic," *Opt.Laser Technol*., Vol.6, pp.391-399, 2000.

[11] H. Huang, M. Ithon, T. Yatagai, and L. Liu," Classified One-Step Modified Signed-Digit Arithmetic and Its Optical Implementation," *Opt.Eng*., Vol.35, No.4, pp.1134-1140, Apr. 1996.

[12] H. Huang, M. Itoh, T. Yatagai, and L. Liu" Classified One-Step Modified Signed-Digit Arithmetic and its Optical Implementation," *Opt.Eng*., Vol.35, No.4, pp.1134-1140, April, 1996.

[13] A. K. Cherri, M. S. Alam, and A. A. S. Awwal," Optoelectronic Symbolic Substitution Based Conical Modified Signed-Digit Arithmetic," *Opt. Laser Technol*., Vol.29, pp.151-157, 1997.

[14] H. Abdeldayem,"Looking for Speed: Go Optical Ultra-Fast Photonic Logic Gates For the Future Optical Communication and Computing," *NASA-Goddard Space Flight Center,* Code 554, Greenbelt, MD 20771, 2003.

[15] N. Savage," SLMs Enter Fields from Photolithography to Optical Computing," *Oemagazine.com/from the magazine/mar03/prodtrends.html*, 2003.